FIG. 1

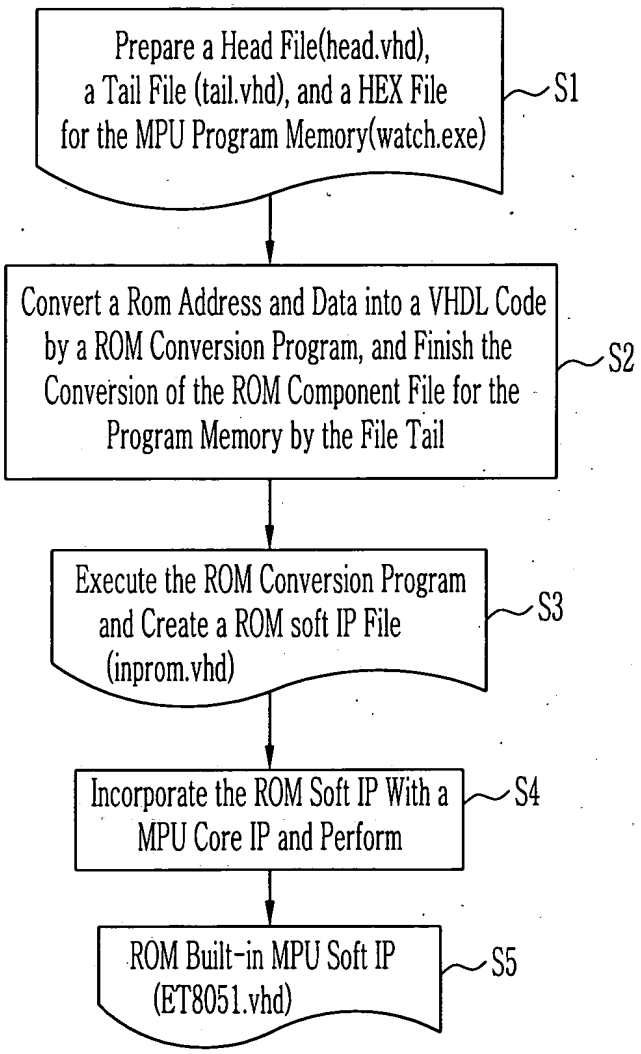

Prepare a Head File(head.vhd), a Tail File (tail.vhd), and a HEX File for the MPU Program Memory(watch.exe)
S1
Convert a Rom Address and Data into a VHDL Code by a ROM Conversion Program, and Finish the Conversion of the ROM Component File for the Program Memory by the File Tail
S2
Execute the ROM Conversion Program and Create a ROM soft IP File (inprom.vhd)
S3
Incorporate the ROM Soft IP With a MPU Core IP and Perform
S4
ROM Built-in MPU Soft IP (ET8051.vhd)
S5

FIG. 2

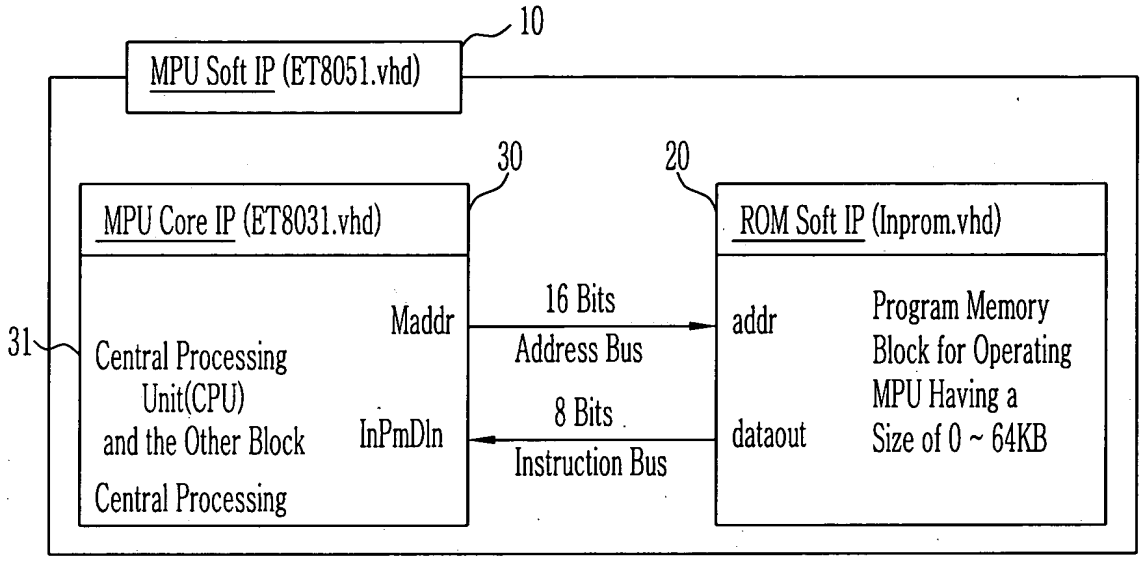

10
MPU Soft IP (ET8051.vhd)
30
20
MPU Core IP (ET8031.vhd)
ROM Soft IP (Inprom.vhd)
31
Central Processing Unit(CPU) and the Other Block
Central Processing
Maddr
InPmDIn
16 Bits
Address Bus
8 Bits
Instruction Bus
addr
dataout
Program Memory Block for Operating MPU Having a Size of 0 ~ 64KB

FIG. 3

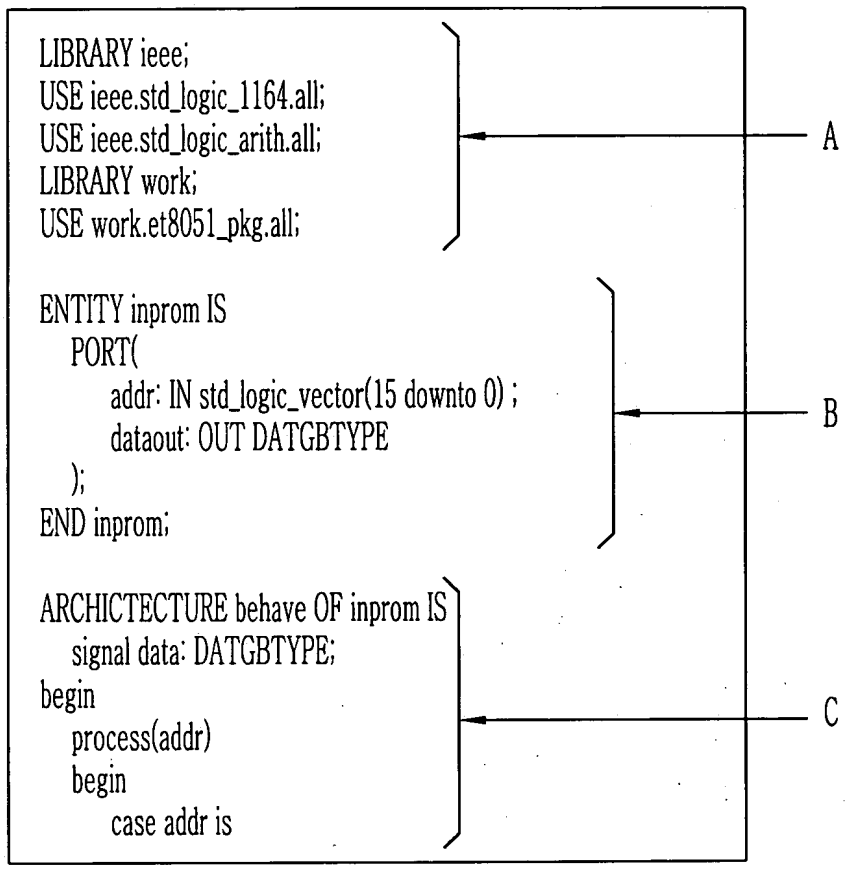

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
LIBRARY work;
USE work.et8051_pkg.all;
ENTITY inprom IS
PORT(
addr: IN std_logic_vector(15 downto 0) ;
dataout: OUT DATGBTYPE
);
END inprom;
ARCHICTECTURE behave OF inprom IS
signal data: DATGBTYPE;
begin
process(addr)
begin
case addr is
A
B
C

FIG. 4

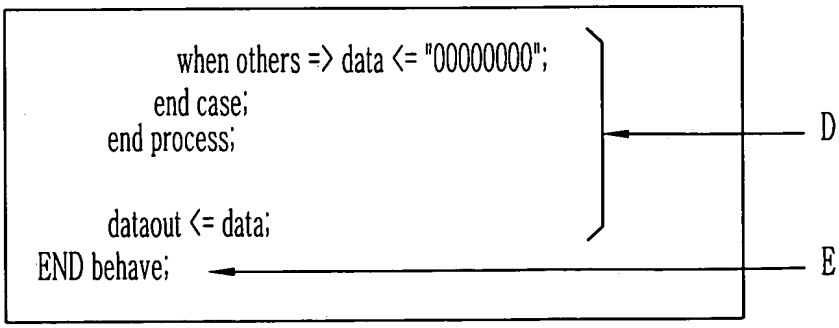

when others => data <= "00000000";
end case;
end process;
dataout <= data;
END behave;
D
E

FIG. 5

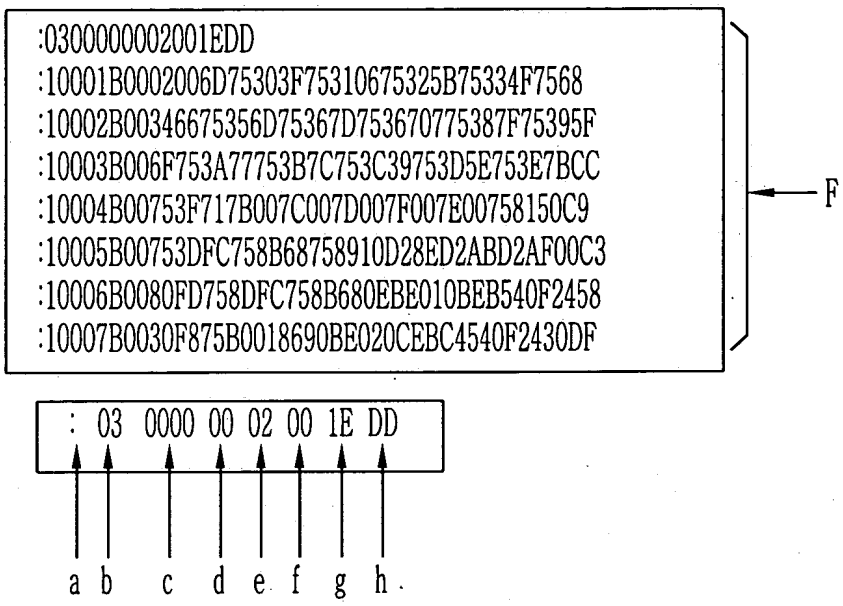

:0300000002001EDD
:10001B0002006D75303F75310675325B75334F7568
:10002B00346675356D75367D753670775387F75395F
:10003B006F753A77753B7C753C39753D5E753E7BCC
:10004B00753F717B007C007D007F007E00758150C9
:10005B00753DFC758B68758910D28ED2ABD2AF00C3
:10006B0080FD758DFC758B680EBE010BEB540F2458
:10007B0030F875B0018690BE020CEBC4540F2430DF
F
: 03 0000 00 02 00 1E DD
a b c d e f g h

FIG. 6A

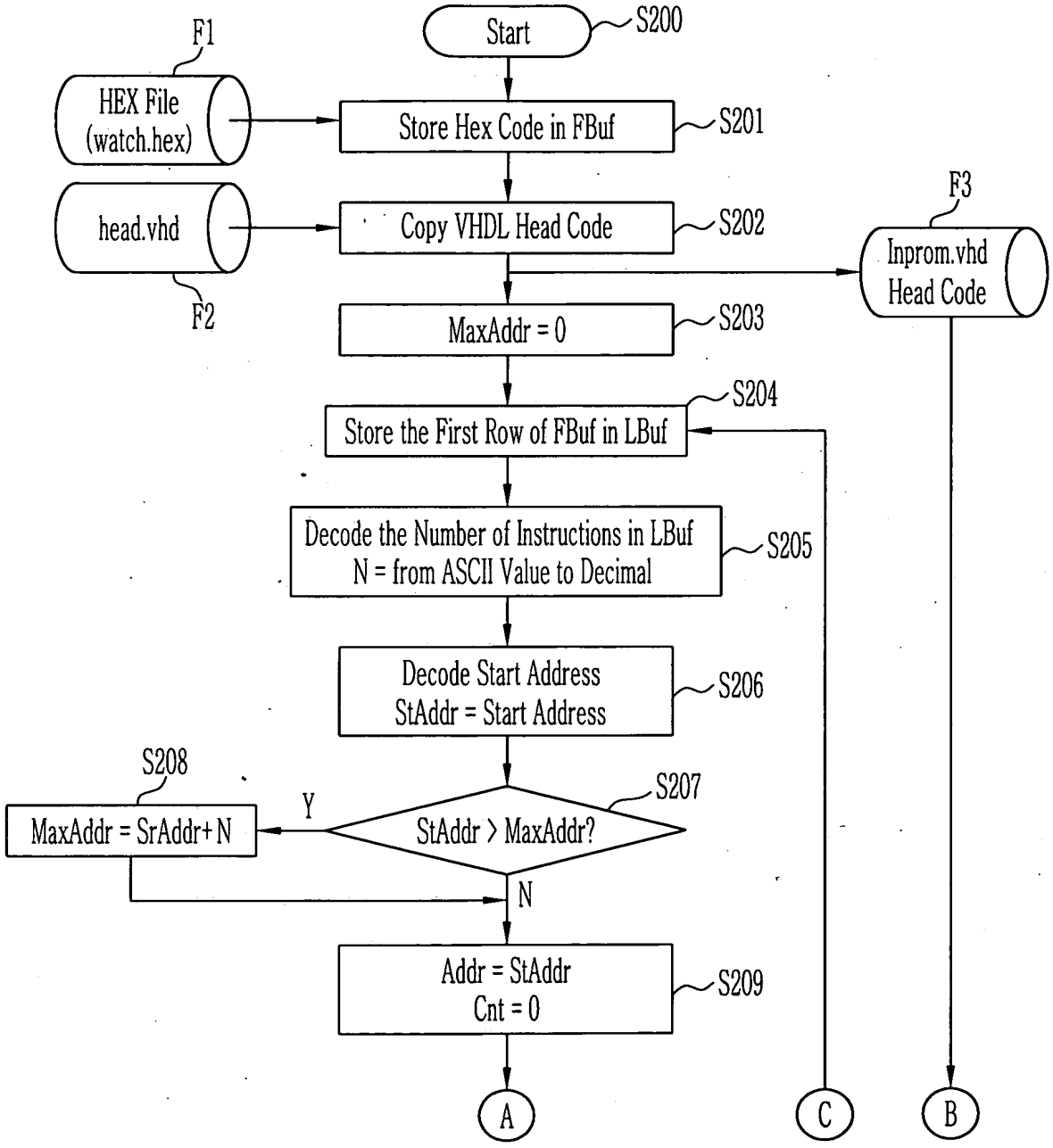

Start
S200
F1
HEX File
(watch.hex)
Store Hex Code in FBuf
S201
head.vhd
F2
Copy VHDL Head Code
S202
F3
Inprom.vhd
Head Code
MaxAddr = 0
S203
S204
Store the First Row of FBuf in LBuf
Decode the Number of Instructions in LBuf
N = from ASCII Value to Decimal
S205
Decode Start Address
StAddr = Start Address
S206
S208
S207
Y
MaxAddr = SrAddr+N
StAddr > MaxAddr?
N
Addr = StAddr
Cnt = 0
S209
A
C
B

FIG. 6B

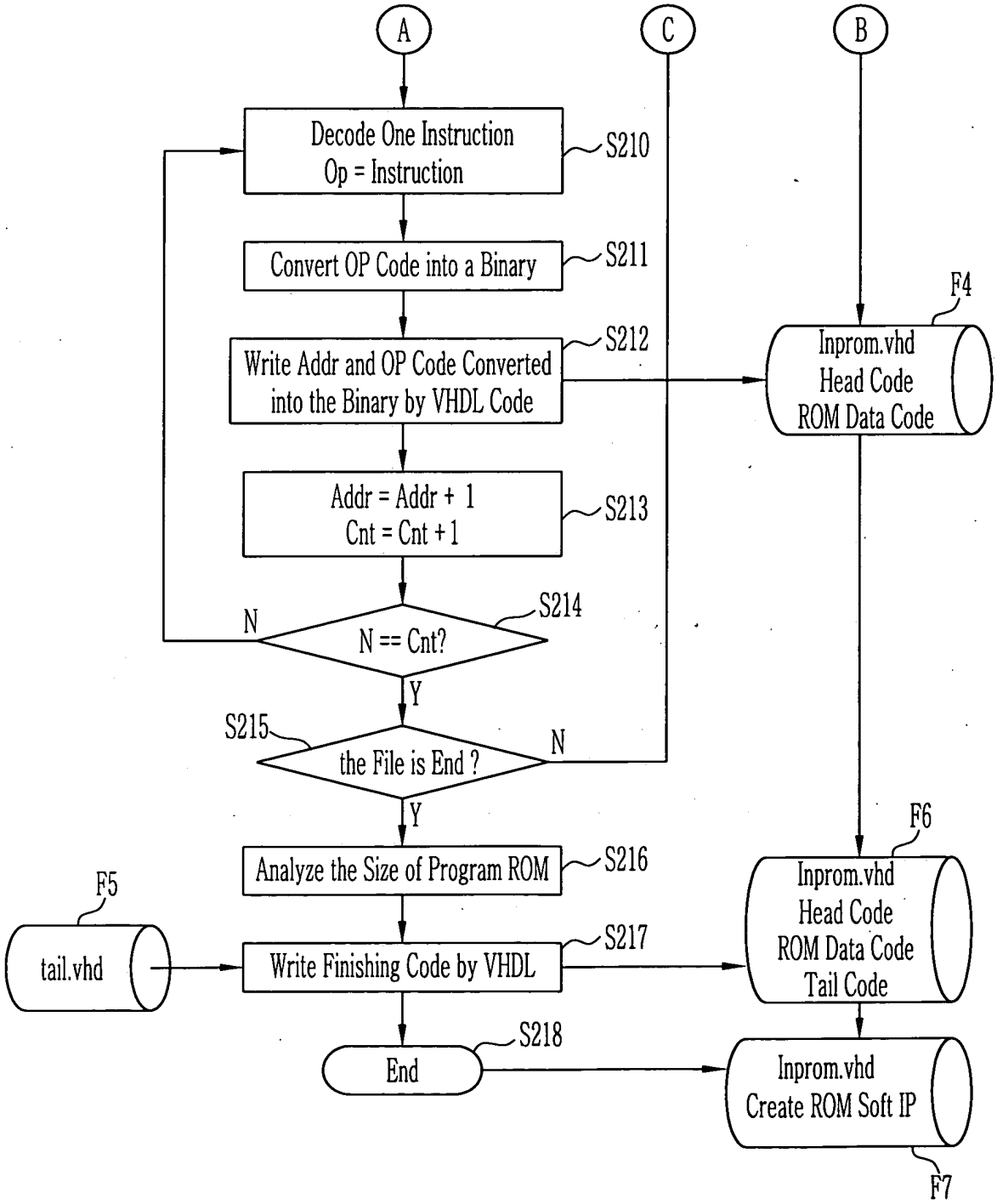
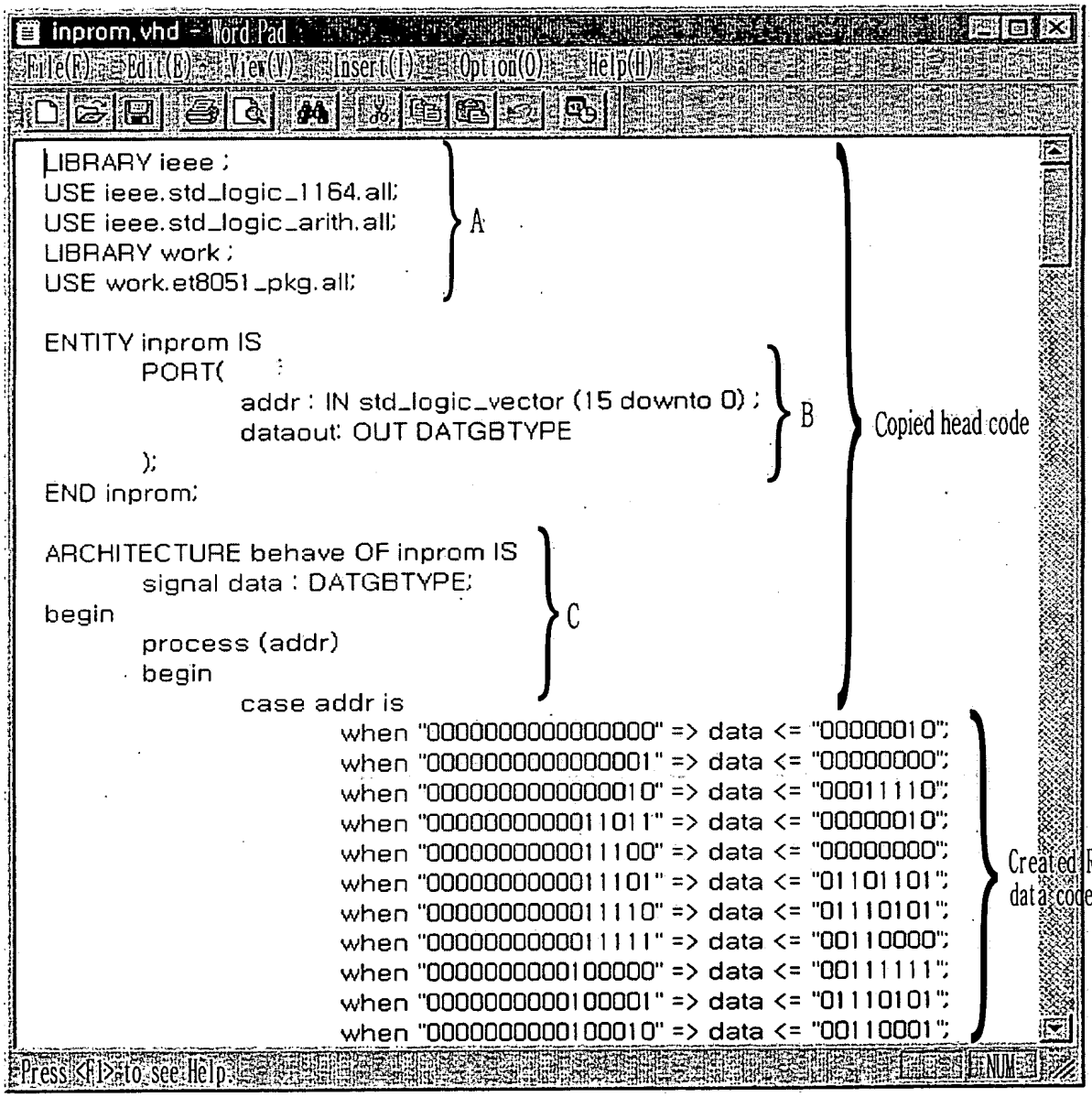

A
C
B
Decode One Instruction
Op = Instruction
S210
Convert OP Code into a Binary
S211
Write Addr and OP Code Converted into the Binary by VHDL Code
S212
F4
Inprom.vhd
Head Code
ROM Data Code
Addr = Addr + 1
Cnt = Cnt + 1
S213
N
S214
N == Cnt?
Y
S215
the File is End ?
N
Y
F6
Analyze the Size of Program ROM
S216
F5
tail.vhd
Write Finishing Code by VHDL
S217
Inprom.vhd
Head Code
ROM Data Code
Tail Code
S218
End
Inprom.vhd
Create ROM Soft IP
F7

FIG. 7


inprom.vhd - Word Pad
File(F) Edit(E) View(V) Insert(I) Option(O) Help(H)
LIBRARY ieee ;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
LIBRARY work ;
USE work.et8051_pkg.all;
A
ENTITY inprom IS
PORT(
addr : IN std_logic_vector (15 downto 0) ;
dataout: OUT DATGBTYPE
);
END inprom;
B
Copied head code
ARCHITECTURE behave OF inprom IS
signal data : DATGBTYPE;
begin
process (addr)
begin
C
case addr is
when "0000000000000000" => data <= "00000010";
when "0000000000000001" => data <= "00000000";
when "0000000000000010" => data <= "00011110";
when "0000000000011011" => data <= "00000010";
when "0000000000011100" => data <= "00000000";
when "0000000000011101" => data <= "01101101";
when "0000000000011110" => data <= "01110101";
when "0000000000011111" => data <= "00110000";
when "0000000000100000" => data <= "00111111";
when "0000000000100001" => data <= "01110101";
when "0000000000100010" => data <= "00110001";
Created ROM data code(F4)
Press <F1> to see Help
NUM